Podmíněné příkazy a bloky

PLIN048 – Základy programování pro humanitní obory

Richard Holaj
FF MU

7. března 2017

Co nás dnes čeká?

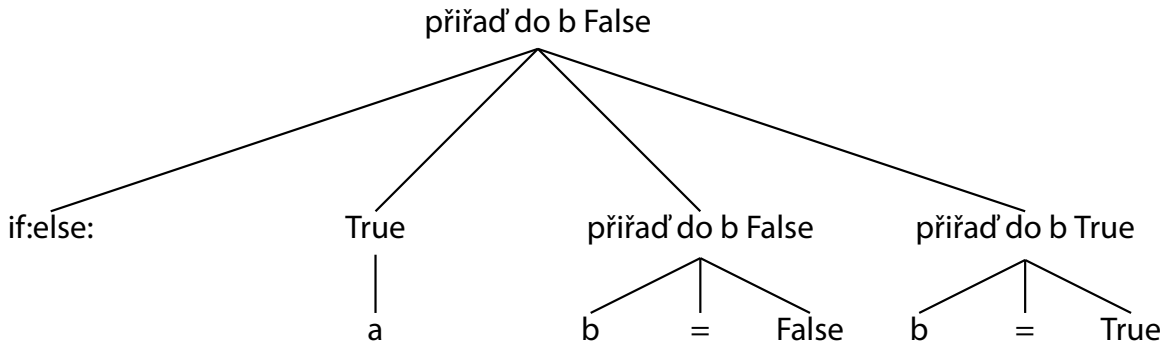
Podmíněné příkazy a větvení programu
- Princip podmínek
- Logické výrazy
- Vícenásobné podmínky a větvení

Bloky kódu
- Blok jako nástroj substituce
- Bloky kódu v různých programovacích jazycích

Cvičení

Princip podmínek

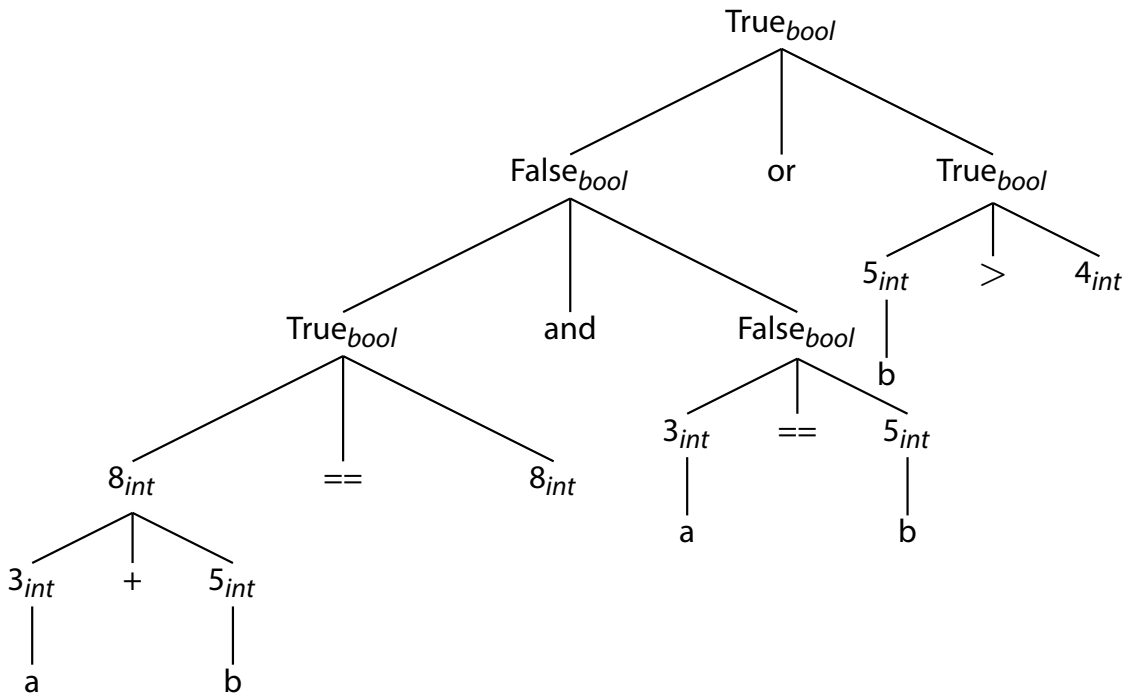
Podmínka v Pythonu




```
1  a = True
2  if a:
3      b = False
4  else:
5      b = True
```

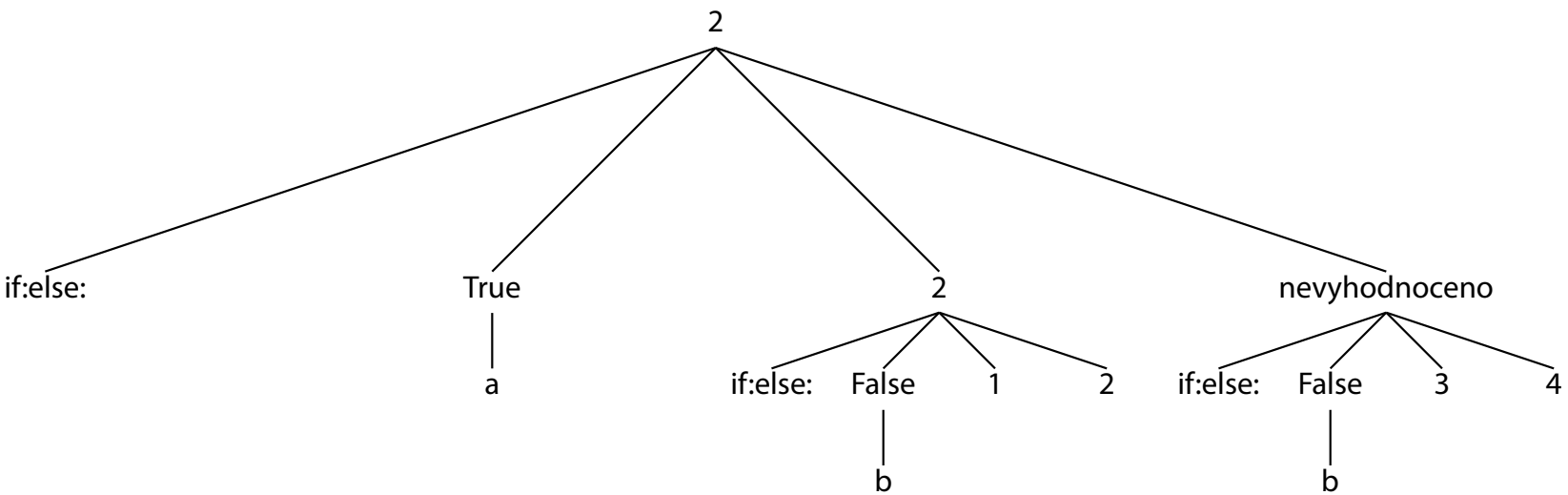
Logické výrazy

Složený logický výraz




```
1 |          a + b == 8 and a == b or b > 4
```

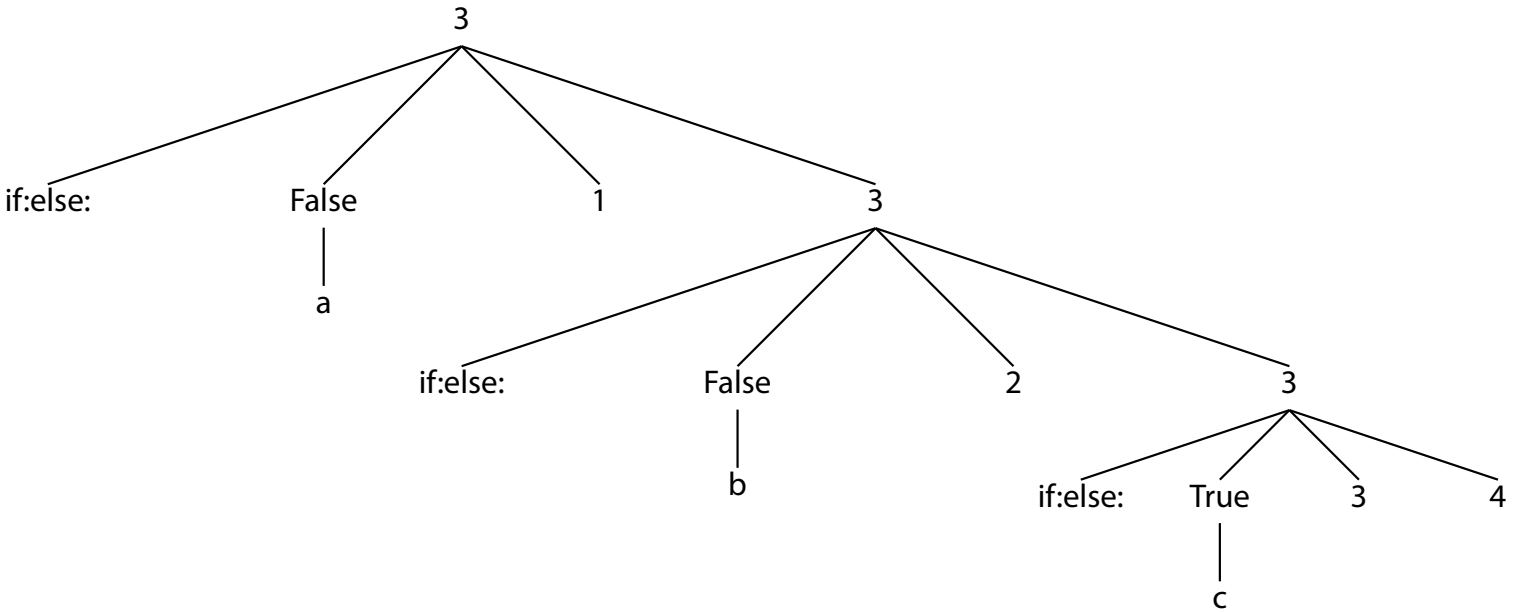
Vícenásobné podmínky a větvení – reprezentace



Vícenásobné podmínky a větvení – zápis

```
1  if a:
2      if b:
3          print(1)     #1
4      else:
5          print(2)     #2
6  else:
7      if b:
8          print(3)     #3
9      else:
10         print(4)     #4
```

Lineárně zanořená podmínka – reprezentace



Lineárně zanořená podmínka – zápis

```
1          if a:
2              print(1)     #1
3          else:
4              if b:
5                  print(2)     #2
6              else:
7                  if c:
8                      print(3)     #3
9                  else:
10                     print(4)     #4
```

```
1          if a:
2              print(1)     #1
3          elif b:
4              print(2)     #2
5          elif c:
6              print(3)     #3
7          else:
8              print(4)     #4
```

Blok jako nástroj substituce

- více příkazů $\rightarrow$ jeden příkaz
- izolované
- neizolované

Bloky kódu v různých programovacích jazycích I

Bloky v JavaScriptu

```
1  ...
2  {
3      var c = 10;
4      console.log(c);
5      {
6          var a = 20;
7          console.log(a);
8      }
9      var d = 30;
10     console.log(d);
11 }
12 ...
```

Bloky kódu v různých programovacích jazycích II

Bloky v Pythonu

```
1 ...
2         c = 10
3         print(c)
4             a = 20
5             print(a)
6         d = 30
7         print(d)
8 ...
```

Bloky kódu v různých programovacích jazycích III

Bloky v Ruby

```
1  ...
2  begin
3      c = 10
4      print(c)
5      begin
6          a = 20
7          print(a)
8      end
9      d = 30
10     print(d)
11 end
12 ...
```

Cvičení

Jaká je hodnota proměnné *c*? *True*, nebo *False*?

```
1 a = 5
2 b = 10
3 c = (a + b > 10 or a != b) and b - a < 3
```

Zvolte hodnoty proměnných *a* a *b* tak, aby platila rovnost na druhém řádku.

```
1 c = a and not b or b and a or b
2 c == False
```

Cvičení

Co vypíše následující kód?

```
1 a = 4
2 b = 7
3 c = 12
4 if a + b < c:
5     print("lower")
6 elif a + b == c:
7     print("equal")
8 else:
9     print("higher")
```

Mějme tři proměnné *a*, *b* a *operation*, přičemž poslední jmenovaná může nabývat hodnot ”+”, ”−”, ”∗” a ”/”. Napište program, který se bude chovat jako jednoduchá kalkulačka vypíše výsledek dle hodnot daných proměnných.

Cvičení

Mějme proměnnou *track_number* **odpovídají aktuálnímu číslu skladby, proměnnou** *track_count* **odpovídající počtu skladeb v playlistu a proměnnou** *repeat* **mající hodnotu** *True* **nebo** *False* **podle toho, zda chceme po přehrání poslední skladby pokračovat od začátku. Napište program odpovídající situaci po přehrání skladby s pořadím odpovídajícím hodnotě proměnné** *track_number*. **Pokud se nejedná o poslední skladbu, přejděte na následující skladbu a vypište, kolikátou skladbu nyní přehráváte, pokud jste na konci, vypište při vypnutém opakování, že přehrávání skončilo, v opačném případě přejděte na první skladbu (opět vypište pořadí).**